Prepared by group 7

Group Project

Uber Booking Management System

16 December 2024

Team Members

Shailja Chaurasia

Tanmay Mehrotra

Nitish Kumar

Anshu Kumar

Introduction

We have covered the key aspects of modern software development, including frontend design, backend setup, API creation, security measures, CI/CD Pipelines and version control workflows, to build a reliable House Sale Price Prediction system..


DianApps
7:28
Pickup spot
Change
Meet the driver at Sherman Circle and Kansas Ave nw
2 min
4.9
6Z47DJ1
Silver Honda Civic
Send a message
Kiley Apartments
UBER
Uber

Problem Statement

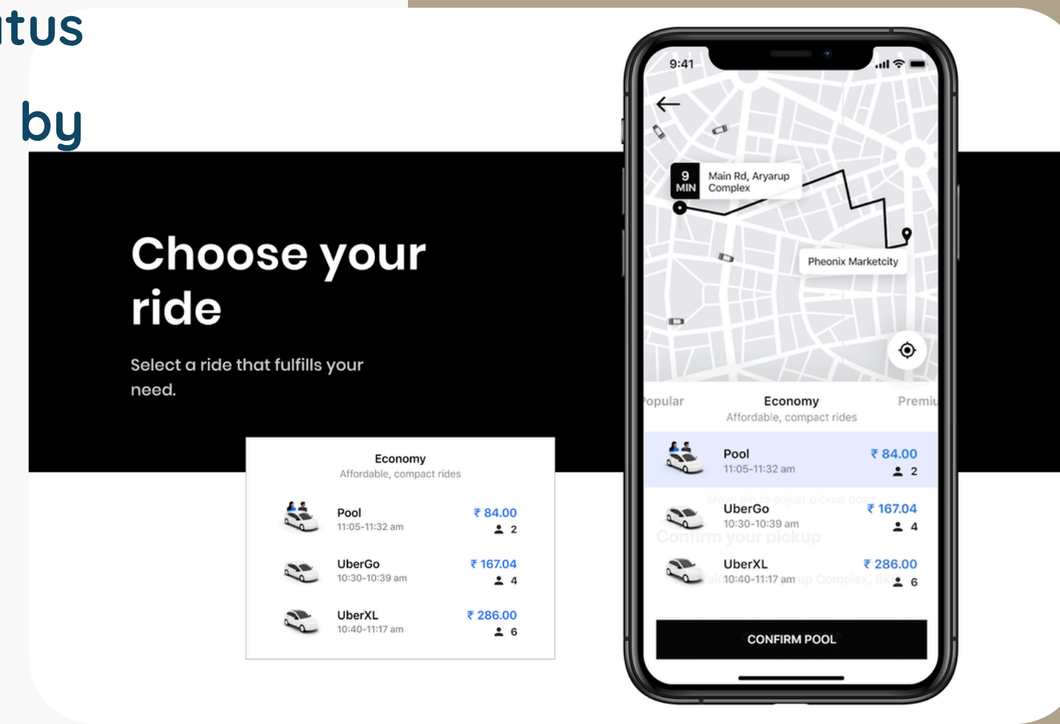
- **Problem: “Manage and optimize Uber ride bookings by predicting ride demand and efficient ride allocations”.**
- **The goal is to improve the experience for both riders and drivers while maximizing platform efficiency and profitability.**

Question: How can ride bookings and allocations be optimized?

Solution: Involves analyzing real-time demand, route optimization, and surge pricing based on factors like – location, time, and availability.

UX Design

- Designed a user-friendly interface for booking management.
- Implemented CRUD operations for rides, users, and driver profiles. like book ride, cancel ride, add ratings
- Integrated features for filtering rides by status (upcoming, ongoing, completed) and sorting data by fare, time, and distance.

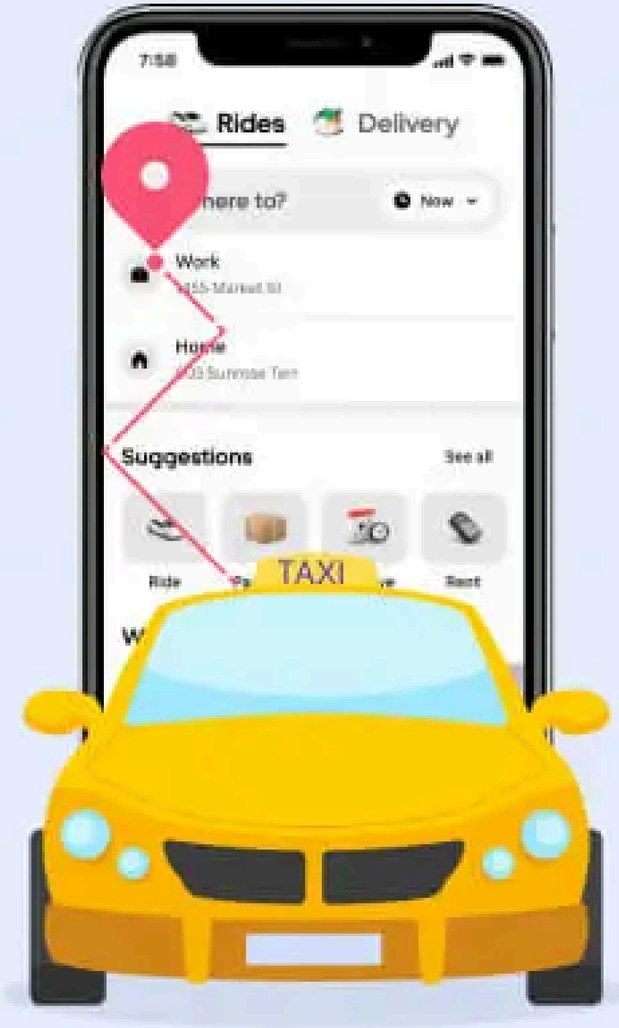


How does Application Work?

1 User Registration

2 Location Sharing

3 Driver Matching



6 Ride Confirmation

5 Ride Execution

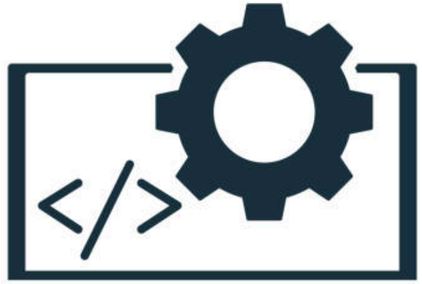
4 Payment Processing

Implementations

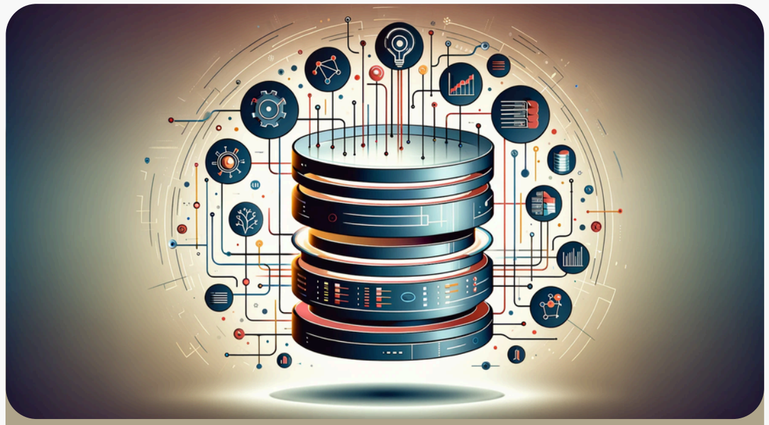
Front End

- Design User Friendly Interface using HTML, CSS, JS, React
- Implement Responsive design using bootstrap

Backend



- choose Mongo DB for storing data
- created efficient schema design
- Implement indexing for faster data retrieval
- Implement API gateway for secure API access
    - Flask , swagger
    - Deployment - Docker

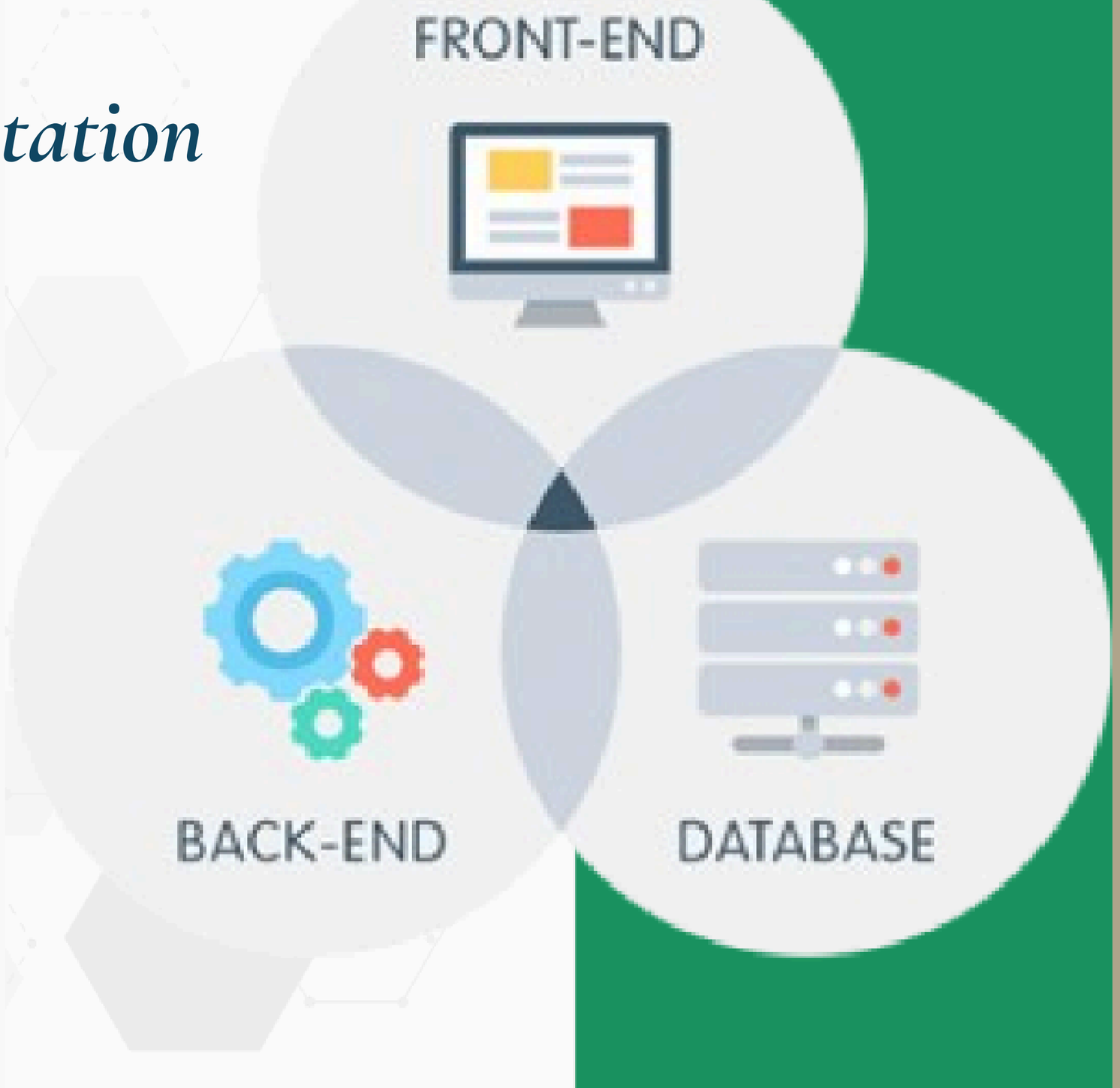


Database

- NOSQL (MongoDB)

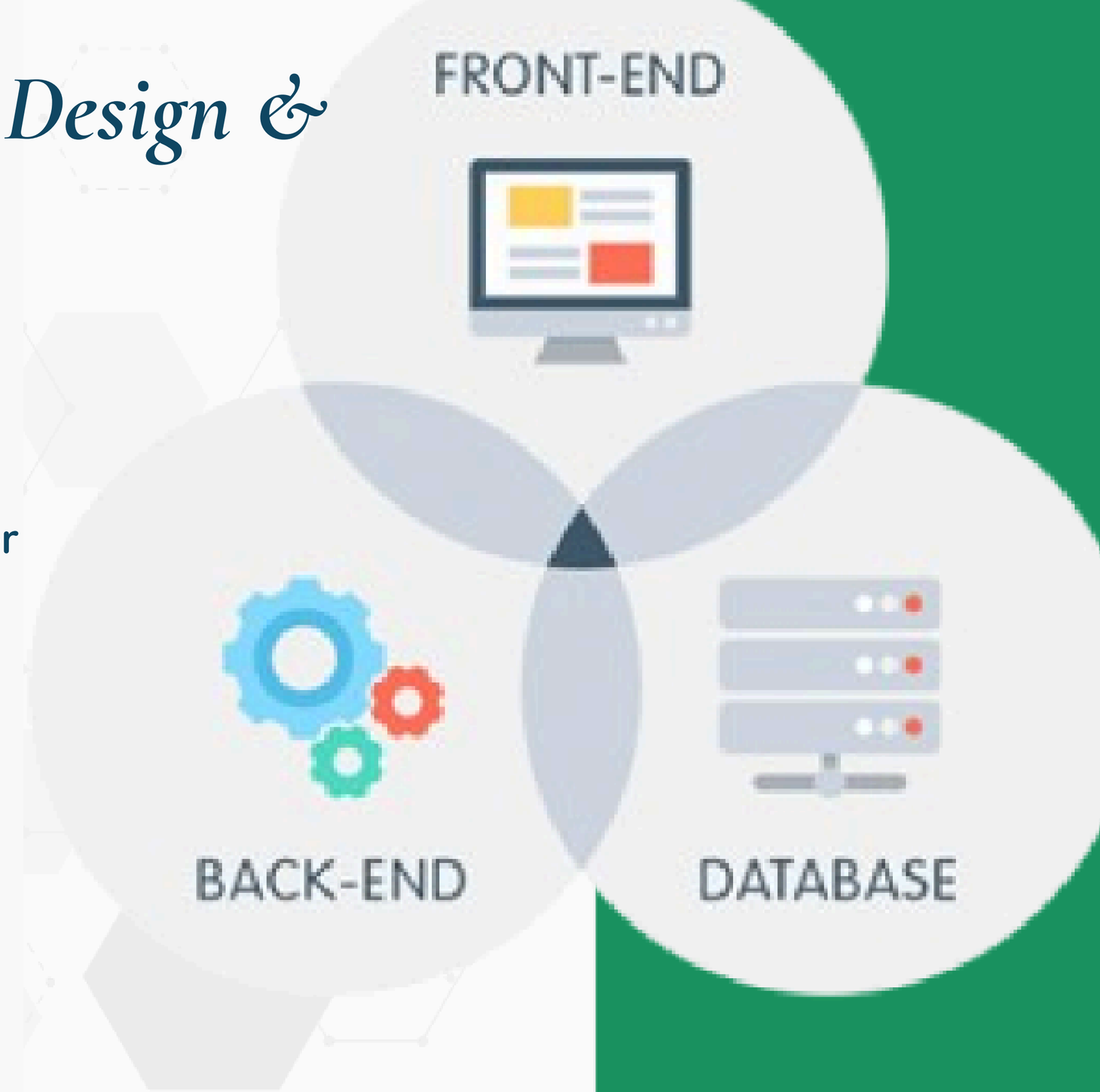
Front End Implementation

- Designed a user-friendly web-Application Interface using HTML, CSS, JavaScript
- Implemented a responsive design for desktop
- Designed Login System
- Used Librarie’s and Framework like React for Building the frontend



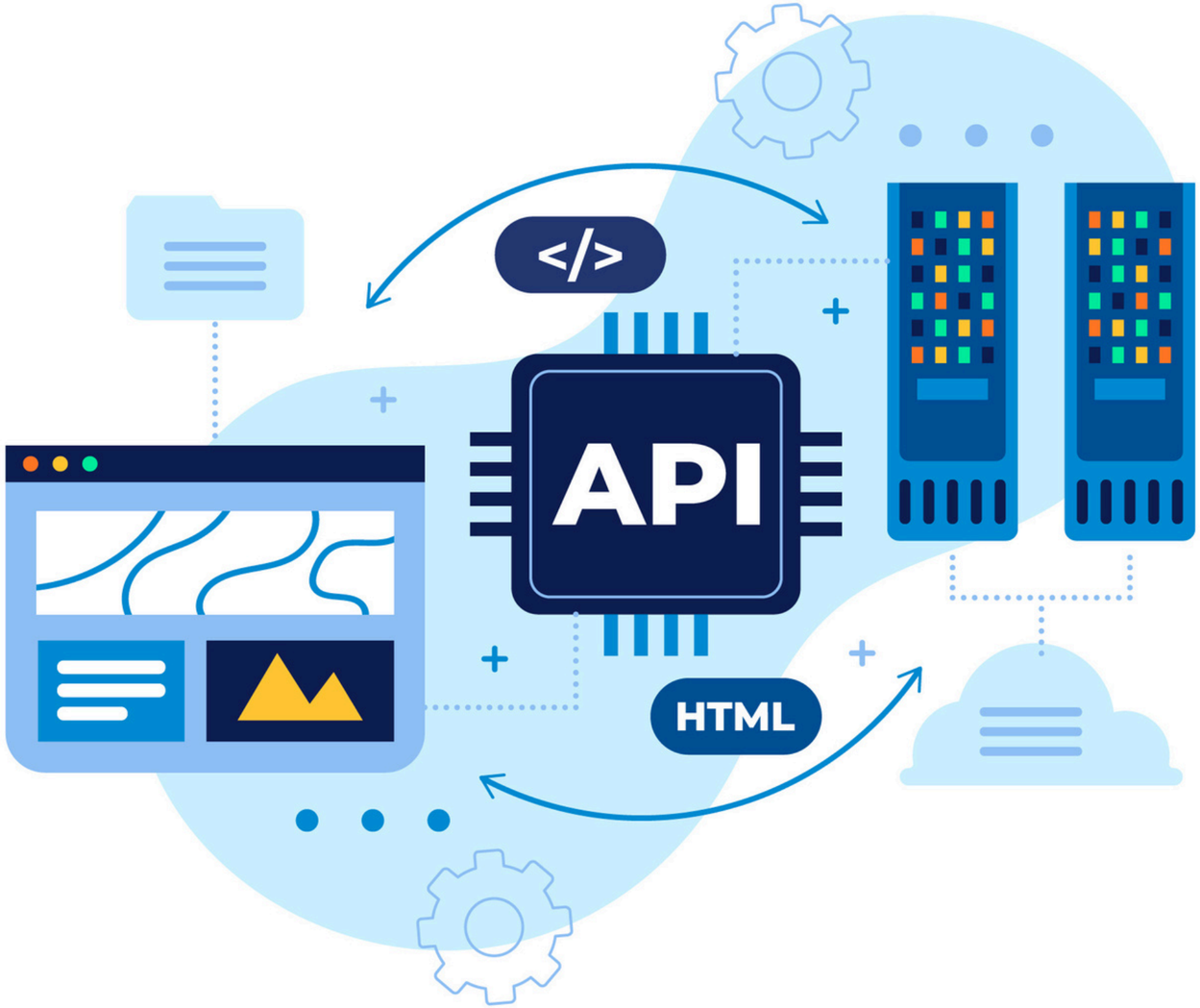
Backend Architecture Design & DataBase

- Choose Database Mongo DB for storing real time User data
- Created an efficient schema for Uber ride Information
- Implemented the Indexing of Database for faster data retrieval
- Implemented an API gateway for Secure API Integration and access



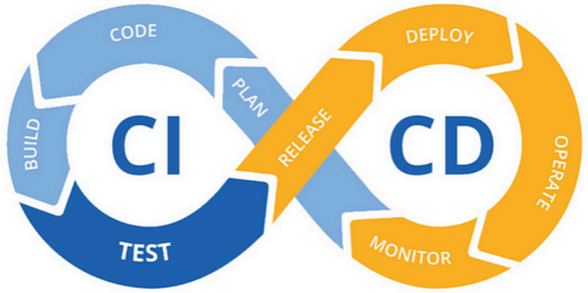
API Design &Integration

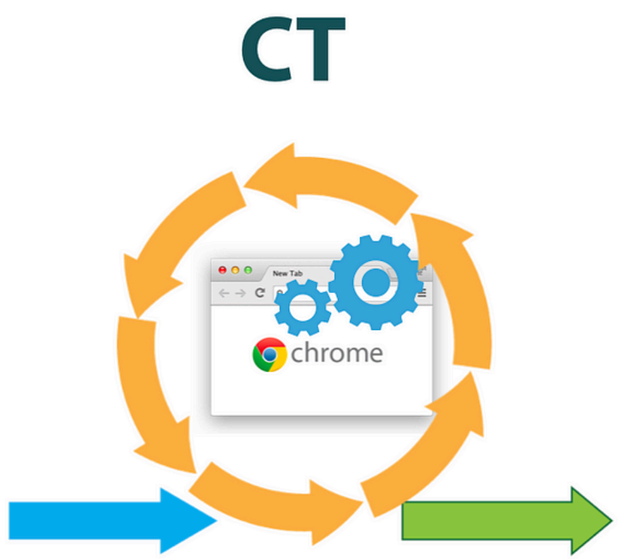
- Includes API endpoints, that is request/response formats, and error handling
- Used API Keys and API Authentication for Securing API access
- Designed a RESTful API for data retrieval and manipulation using Flask



CI/CD Pipelines

- Used GitLab CI/CD to automate build, test, and deployment processes.
- Configured pipelines for testing
- Setup environment-specific deployment stages (dev, staging, production).
- Utilized caching in pipelines to optimize build and test execution times.
- application Used Dockerization for containerization and deployment





GitLab Branching

- Used a CI/CD tool like GitLab CI/CD
- used feature branches for development and testing
- used merge request for code reviews and approval
- used the main branch for production deployment

Result

Performance Optimization & Security Measures

For Performance Optimization

- Implemented Code Splitting, lazy loading and efficient state management.
- Optimized API and database queries.
- Used CDNs for static asset delivery. Monitored real-time performance metrics

For Security Measures

- Conducted regular Vurnerability assessments.
- Implemented secure coding practices, HTTPS, and proper authentication.
- Kept dependencies updated and used security monitoring tools

Continuous Monitoring

Set up alert for performance degradation and security issues regular review of logs and security scans in CI/CD piplines

Set up for User data Protection

- Implemented encryption into login system for sensitive data at rest.
- In transit followed data protection regulations like GDPR.
- Regular backup and test data recovery procedure

Challenges Faced

1. User Interface (UI) and Experience (UX):

- Challenge: Designing a simple, intuitive, and effective interface for both riders is a key challenge.
- Solution: Focus on simplicity, providing helpful on-screen instructions, easy navigation, and considering user feedback to iteratively improve the design.

2. Payment Gateway Integration

- Challenge: Integrating secure and reliable payment gateways (e.g., PayPal, Stripe, etc.) to handle transactions efficiently and safely.
- Solution: Choosing a reliable and well-documented payment gateway, ensuring compliance with PCI-DSS standards, and testing the payment flow thoroughly can ensure a smooth payment process.

3. Maintaining System Reliability

Utilizing distributed architectures, conducting regular load tests, monitoring system performance

4. Data Security and Privacy

Collecting personal and payment information from users strong encryption for data storage and transmission

location and Mapping

Conclusion

Our Fullstack Uber booking management system provides a dynamic, real-time solution for ride booking and management. It streamlines operations for riders and drivers, ensuring seamless bookings, efficient ride allocation, and accurate trip management. Designed with scalability, security, and performance at its core, the system ensures fast and reliable processing as user demand grows.

Thank you